The Status of the Claims

1. (Currently Amended) A method comprising:

~~utilizing~~ placing a message from a first virtual machine associated with a first virtual machine queue into a second virtual machine queue associated with a ~~and~~ second virtual machine, ~~queues associated with respective first and second virtual machines to communicate between the virtual machines using data stored in a page in response to an instruction,~~ wherein the message is associated with a page of a first address space of the first virtual machine;

~~wherein utilizing comprises:~~

mapping the page of the first address space ~~communicating the data stored in the page from the first virtual machine~~ to be associated with a second address space of the second virtual machine by updating a page table, via [[by]] a processor; and ~~by remapping the page from being associated with the first virtual machine to being associated with the second virtual machine; and~~

causing the second virtual machine to process the message.

~~wherein said updating comprises:~~

~~determining whether the instruction requires an immediate VM exit from the first virtual machine; and~~

~~exiting the first virtual machine based on said determining.~~

2. (Cancelled).

3.   (Currently Amended) The method according to claim 1, further comprising ~~wherein updating further comprises:~~

~~placing at least one of data or an address associated with the page into a first virtual machine control structure associated with the first virtual machine;~~

~~placing the at least one of data or address into the second virtual machine queue; and~~

dequeueing the second virtual machine queue.

4.   (Currently Amended) The method according to claim 3, wherein dequeueing includes~~:~~

~~reading the at least one of data or address into a second virtual machine control structure associated with the second virtual machine; and~~

storing ~~the at least one of data or address into the~~ an address ~~space~~ associated with the page into a local variable associated with the second virtual machine.

5.   (Currently Amended) The method according to claim [[3]] 1, further comprising ~~wherein the page contains a message and the method further comprises:~~ ~~processing the message within the second virtual machine~~ determining whether a write instruction used to place the message in the second virtual machine queue involves an immediate virtual machine exit.

6.   (Currently Amended) The method according to claim [[3]] 5, ~~wherein exiting~~
~~occurs immediately after placing the at least one of data or an address associated with the~~
~~page into the first virtual machine control structure~~ further comprising, when the write
instruction involves an immediate virtual machine exit, causing an immediate virtual machine
exit such that a virtual machine monitor can process the message, and, when the write
instruction does not involve an immediate virtual machine exit, processing the message in
response to a natural virtual machine exit.

7.   (Original) The method according to claim 1, further comprising:

conveying identification information associated with the first and second virtual
machines between the first and second virtual machines via the first and second virtual
machine queues.

8.   (Currently Amended) A computer system comprising:

at least one hardware processor; and

a computer readable memory comprising program instructions, executable by the at least one processor, for:

~~first and second virtual machines;~~

a first virtual machine control structure associated with [[the]] a first virtual machine, the first virtual machine control structure having a first virtual machine queue adapted to enqueue and dequeue a message;

a second virtual machine control structure associated with [[the]] a second virtual machine, the second virtual machine control structure having a second virtual machine queue adapted to enqueue and dequeue a message, the second virtual machine queue to receive a message from the first virtual machine, wherein the message is associated with a page of a first address space of the first virtual machine; and

a virtual machine monitor ~~coupled~~ to ~~the first and second virtual machines and to the first and second virtual machine control structures, the virtual machine monitor adapted to supervise communication between the first and second virtual machines and, in response to an instruction, communicate a message stored in a~~ map the page of the first address space ~~from the first virtual machine to the second virtual machine by update of a page table by a processor by remap of a page from being~~ to be associated with a second address space of the ~~first~~ second virtual machine by updating a page table ~~to being associated with the second virtual machine, the update comprising :~~, the virtual machine monitor to cause ~~exit~~ the ~~first~~ second virtual machine ~~based on a determination whether the instruction requires an immediate VM exit from the first virtual machine~~ to process the message.

9.   (Cancelled)

10.   (Currently Amended) The computer system according to claim 8, wherein the first virtual machine ~~monitor~~ is ~~further adapted~~ to receive a second message from the second virtual machine, wherein the message is ~~place at least one of data or an address~~ associated with [[the]] a second page of a second address space of ~~into~~ the ~~first~~ second virtual machine ~~control structure~~.

11.   (Cancelled).

12.   (Currently Amended) The computer system according to claim [[10]] 8, wherein the first virtual machine ~~monitor~~ is ~~further adapted~~ to determine whether a write instruction used to place ~~at least one of data and or an address into~~ the message into the second virtual machine queue involves an immediate virtual machine exit.

13.   (Currently Amended) The computer system according to claim 12, wherein the ~~second~~ first virtual machine undergoes an ~~is adapted to process the page~~ immediate VM exit in response to determining that the write instruction involves an immediate exit, and wherein the virtual machine monitor processes the message in response to a natural exit when the write instruction does not involve an immediate exit.

14.   (Original) The computer system according to claim 8, wherein the virtual machine monitor is further adapted to convey identification information associated with the first and second virtual machines between the first and second virtual machines via the first and second virtual machine queues.

15.   (Currently Amended) A computer readable memory ~~containing program~~ having instructions stored thereon that, when executed ~~by a processor~~, cause ~~the processor~~ a machine to:

~~utilize~~ place a message from a first ~~and second~~ virtual machine associated with a first virtual machine queue into a second virtual machine queue ~~queues~~ associated with ~~respective first and~~ a second virtual ~~machines~~ machine ~~to communicate between the virtual machines using data stored in a page in response to an instruction~~, wherein the message is associated with a page of a first address space of the first virtual machine;

map the page of the first address space ~~wherein utilize comprises communicate the data stored in the page from the first virtual machine~~ to be associated with a second address space of the second virtual machine by ~~update~~ updating [[of]] a page table ~~by a processor by remap of the page from being associated with the first virtual machine to being associated with the second virtual machine~~; and

cause the second virtual machine to process the message.

~~wherein the update comprises:~~

~~exit from the first virtual machine based on a determination whether the instruction requires an immediate VM exit from the first virtual machine.~~

16.   (Cancelled).

17.   (Currently Amended) The computer readable memory according to claim 15, ~~containing further program~~ having instructions stored thereon that, when executed ~~by a processor~~, cause ~~the processor~~ a machine to:

~~place at least one of data or an address associated with the page into a first virtual machine control structure associated with the first virtual machine;~~

~~place the at least one of data or address into the second virtual machine queue; and~~

dequeue the second virtual machine queue.

18.   (Currently Amended) The computer readable memory according to claim [[15]] 17, ~~containing further program instructions that, when executed by a processor, cause the processor to:~~

~~read the at least one of data or address into a second virtual machine control structure associated with the second virtual machine; and~~ wherein dequeuing the second virtual machine queue includes

~~store~~ storing ~~the at least one of data or address into the~~ an address ~~space~~ associated with the page into a local variable associated with the second virtual machine.

19.   (Currently Amended) The computer readable memory according to claim

[[17]] 15, ~~wherein the page contains a message, and wherein the computer readable memory~~

~~contains further program~~ having instructions stored thereon that, when executed ~~by a~~

~~processor~~, cause ~~the processor~~ a machine to:

~~process the message within the second virtual machine~~ determine whether a write

instruction used to place the message in the second virtual machine queue involves an

immediate virtual machine exit.

20.   (Currently Amended) The computer readable memory according to claim

[[15]] 19, ~~containing further program~~ having instructions stored thereon that, when executed

~~by a processor~~, cause ~~the processor~~ a machine to:

when the write instruction involves an immediate virtual machine exit, cause an

immediate virtual machine exit such that a virtual machine monitor can process the message;

and

when the write instruction does not involve an immediate virtual machine exit,

process the message in response to a natural virtual machine exit.

~~convey identification information associated with the first and second virtual~~

~~machines between the first and second virtual machines via the first and second virtual~~

~~machine queues.~~

21.   (Cancelled).

22.   (Cancelled).